Investigating the orthogonality of model compression by pruning and knowledge distillation.

Roger Garcia Caño

June 30, 2025

Abstract

This work investigates the interaction between two widely adopted model compression techniques —pruning and knowledge distillation (KD)— to determine whether their effects are orthogonal, overlapping, or interfering. While both methods independently reduce the size of overparameterized neural networks while preserving performance, their combined application has not been systematically analyzed. We hypothesize that pruning and KD may target similar representational subspaces, potentially limiting their additive benefits. To explore this, we conduct a series of controlled experiments applying various pruning ratios and distillation setups, evaluating their individual and joint impact on model accuracy, robustness, and representational structure. Through representation similarity analysis using techniques such as Centered Kernel Alignment (CKA), we assess whether pruned and distilled models converge to analogous feature spaces. Our findings aim to clarify whether these compression strategies are complementary or redundant, providing guidance on how to effectively combine them in practice.

Keywords: neural network compression, pruning, knowledge distillation, orthogonality, representational similarity, Centered Kernel Alignment (CKA), model efficiency, deep learning, subnetwork discovery, information bottleneck

1 Introducction - Context of the Project

Deep neural networks have achieved remarkable success across a wide range of tasks, often at the cost of significant computational and memory requirements. This overparameterization not only increases the resources needed for training and inference but also limits their deployment. Two prominent strategies to address these challenges are network pruning, the removal of redundant parameters, and knowledge distillation, the transfer of knowledge from a large teacher model to a smaller student model.

While both pruning and distillation have been extensively studied independently, their combined or sequential application remains an active area of research. Understanding how these techniques interact, and how the order and degree of pruning or distillation affect model performance, is crucial for designing efficient and robust neural networks.

In this work, we explore the relationship between pruning and distillation. We propose and analyze several strategies: (1) iterative pruning with fine-tuning, (2) distillation at various pruning levels, and (3) hybrid approaches where distillation and pruning are combined. By visualizing the performance trajectories across different pruning and distillation schedules, we provide insights into the trade-offs and combined effects between these methods. Our results aim to guide practitioners in selecting effective compression techniques and pipelines for deploying deep learning models in resource-constrained environments.

• Contact E-mail: rogergarciacanyo@gmail.com
• Supervised by: Andrey Barsky (Computer Science)
• Academic Year 2024/25

2 State of the Art

2.1 Neural Network Compression Techniques

Neural network compression has become a central topic in deep learning, with pruning and knowledge distillation emerging as two of the most prominent strategies. Magnitude-based pruning was first introduced by LeCun et al., who proposed the Optimal Brain Damage method for removing less important weights [15]. This idea was later extended by Han et al., who demonstrated that iterative pruning and fine-tuning could eliminate up to 90% of a network's parameters while maintaining accuracy [23]. As research progressed, the distinction between structured and unstructured pruning became important. Li et al. showed that structured pruning, which removes entire filters or channels, is more compatible with hardware acceleration [24], while Liu et al. proposed channel pruning

to achieve practical speedups on standard hardware [25]. The Lottery Ticket Hypothesis, introduced by Frankle and Carbin, further advanced the field by revealing that sparse subnetworks within dense models can be trained to reach comparable accuracy as the original network [13].

Knowledge distillation, on the other hand, was formalized by Hinton et al., who showed that a smaller student network could learn to mimic the soft outputs of a larger teacher, capturing valuable information about class relationships [14]. Beyond this foundational approach, Yim et al. introduced feature-based distillation methods such as Attention Transfer and Flow of Solution Procedure, which transfer intermediate feature relationships between teacher and student [26]. Self-distillation has also gained attention, with Kim et al. demonstrating that a network can improve by distilling from itself [27].

2.2 Combined Compression Approaches

Recent research has increasingly focused on combining pruning and knowledge distillation, either sequentially or in hybrid frameworks. One common approach is to apply pruning first, followed by distillation. Aghli et al. systematically studied this sequence, finding that knowledge distillation can help recover performance lost during aggressive pruning [1]. Park et al. further demonstrated that this order is particularly effective at high sparsity levels [6]. Conversely, Zhu and Gupta explored the reverse order, showing that networks distilled before pruning are more robust to subsequent parameter removal [22]. Hybrid and iterative methods, such as those proposed by Muralidharan et al. and Pan et al., alternate between pruning and distillation to maximize both compression and accuracy [4, 5].

2.3 Representation Analysis and Orthogonality

Understanding how compression techniques affect internal representations has become a key research direction. Kornblith et al. established Centered Kernel Alignment (CKA) as a robust metric for comparing neural representations, showing its invariance to orthogonal transformations and its ability to capture non-linear relationships [10]. Further studies by Davari et al. linked representation similarity to transfer learning performance and reliability [9]. The question of whether pruning and distillation act orthogonally or redundantly has been explored in previous work, but often without a deep analysis of their interaction. Many studies assume that combining both techniques results in additive gains, yet this assumption has not been verified. Studies by Aghli and Ribeiro, as well as Park and No, demonstrated that combining pruning and distillation may yield additive benefits, suggesting partially orthogonal mechanisms [1, 6].

2.4 Research Gaps and Positioning

Despite these advances, most prior work has focused on empirical combinations of pruning and distillation, with limited systematic analysis of their representational orthogonality. This thesis addresses this gap by employing CKA-based analysis across multiple compression levels, aiming to clarify when pruning and distillation are complementary and when they are redundant. By providing an exhaustive, layer-wise analysis of compressed models, this work offers practical insights for selecting optimal compression strategies in resource-constrained environments.

3 Objectives

The objectives for this work are divided into two main components:

Orthogonality Assessment

The goal of this first objective is to evaluate the degree of independence (orthogonality) between pruning and knowledge distillation. More specific:

- Analyze the effect of applying pruning and knowledge distillation individually and in combination, considering both performance and the overall similarity of internal representations produced by compressed models.
- Determine if the improvements gained with both techniques are additive, collaborative, or interfering.
- Compare the results with baseline models to contextualize the observed effects.

Substructure Analysis

This second component focuses on understanding and analyzing the structural implications of each compression method:

- Investigate if pruning and knowledge distillation identify and preserve similar or distinct substructures from the original model.
- Compare the activations that models compressed with different techniques can have to same input samples.
- Analyze how the combination of these methods influences the diversity or overlap of preserved substructures across different scenarios.

4 Theoretical Foundations for Non-Orthogonality Between Pruning and Distillation

Understanding why pruning and knowledge distillation may not act as fully independent (orthogonal) compression strategies requires a look at several theoretical perspectives from deep learning. In this section, we review key concepts that together provide a foundation for formulating such a hypothesis. These frameworks help explain why, under certain conditions, pruning and distillation may guide neural networks toward similar solutions, despite their different mechanisms.

4.1 The Manifold Hypothesis and Representation Convergence

The Manifold Hypothesis suggests that high-dimensional data (like images) lies on or near lower-dimensional manifolds embedded in the high-dimensional space. Neural

networks learn to map inputs to these underlying manifolds through their hierarchical representations.

Both pruning and distillation are basically manifold compression techniques that may converge to similar underlying geometric structures, compromising the assumption of orthogonality.

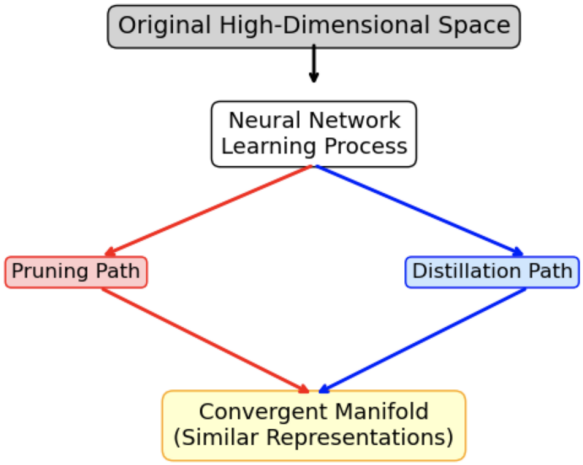

Fig. 1: Manifold Convergence Theory.

4.2 Information Bottleneck Theory

This principle (Tishby & Zaslavsky, 2015) [20], provides a framework to analyze the trade-off between compression and prediction quality in neural networks. In the information plane diagram (Fig. 2), this trade-off is represented between:

- R: the mutual information between the input X and the internal representation T.
- D_{IB}: the IB distortion (mutual information loss with the target Y).

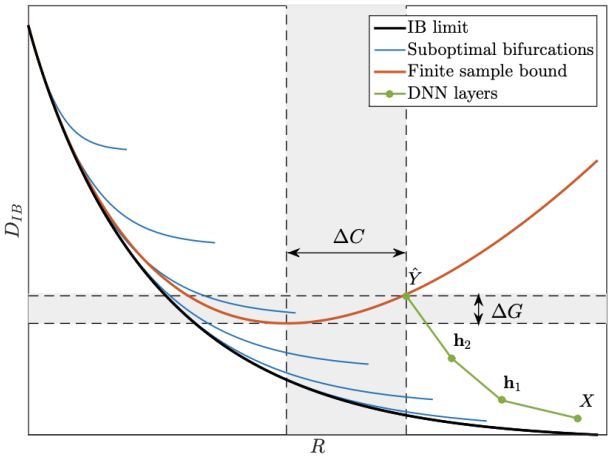

Fig. 2: Information Bottleneck Principle.

IB cruve explanation:

- The black curve is the theoretical D_{IB} limit, which defines the optimal trade-off: the minimal distortion D_{IB} for a given compression R.
- The red curve is a finite sample bound, representing realistic generalization errors due to limited data.
- The green path shows how a deep neural network's layers progress through this space during training, starting at the input X and ending at the representation closest to predicting Y.
- The blue lines are suboptimal trajectories caused by design or training inefficiencies.

Distillation aims to guide a student model to replicate a teacher model's internal representations or outputs. This can be seen as an attempt to move the green dots (layer representations) downward, closer to the IB curve, improving generalization by reducing the gap ΔG in the figure.

Pruning, on the other hand, reduces model complexity—effectively a movement leftward along the R-axis (compression). But if done too aggressively or without guidance, it can lead to increased distortion D_{IB} and exceed the finite sample bound, resulting in worse generalization.

Why non-orthogonal:

- Distillation implicitly helps compression (a goal of pruning) by regularizing representations and reducing variance, reducinf the network's effective information flow.
- Pruning, especially structured pruning, alters internal representations, which impacts how well the model retains and transmits information about Y, which is exactly what distillation is trying to control.

The relation between pruning and distillation might not be orthogonal because both techniques modify the model's internal representations with respect to the compression–generalization trade-off. The IB principle makes this visible: both techniques aim to reposition the layers (green points) in the information plane, through different mechanisms.

4.3 Loss Landscape Geometry

Recent research shows that *sublevel sets of the loss surfaces of overparameterized networks are connected, exactly or approximately* [11].

This suggests that different compression techniques may explore the same underlying loss regions.

So, pruning and knowledge distillation may discover different paths to the same optimal regions in parameter space, resulting in functionally similar representations despite different compression mechanisms.

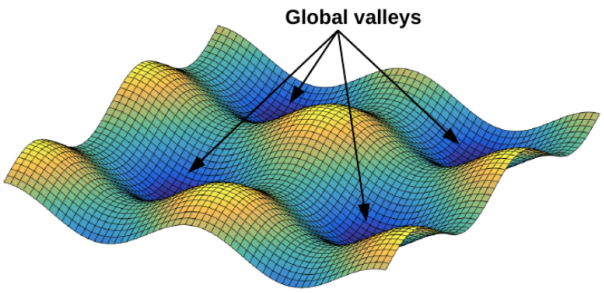

Fig. 3: Illustration of a loss landscape with multiple global valleys. The absence of bad local valleys suggests that different optimization paths (e.g., pruning or distillation) can independently reach similar low-loss regions due to the connectedness of sublevel sets.

4.4 Lottery Ticket Hypothesis and Subnetwork Discovery

The Lottery Ticket Hypothesis (Frankle & Carbin, 2019) [13] demonstrates that sparse subnetworks can achieve comparable performance to full networks when trained in isolation.

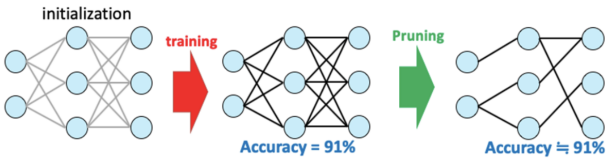

Fig. 4: Lottery ticket hypothesis

This suggests that:

- Optimal subnetworks exist within larger networks
- Multiple compression techniques may discover similar or overlapping subnetworks.
- Knowledge distillation and Pruning may implicitly guide networks toward these same optimal substructures

4.5 Universal Approximation and Capacity Constraints

The Universal Approximation Theorem states that a feed-forward neural network with at least one hidden layer and a suitable activation function (such as sigmoid or ReLU) can approximate any continuous function on a compact subset of R^n, given sufficient width (number of neurons) [**?**, **?**]. Formally, for any continuous function $f : R^n \to R$ and any $\epsilon > 0$, there exists a neural network $\hat{f}$ such that

$$\sup_{x \in K} |f(x) - \hat{f}(x)| < \epsilon,$$

where K is a compact subset of R^n.

However, in practice, the capacity of a network is limited by the number of parameters, especially after compression via pruning or distillation. When capacity is constrained, the set of functions that the network can represent is narrowed, and the network is forced to approximate the target function as closely as possible within these limits. This restriction can lead to a situation where different compression techniques (such as pruning and distillation) converge to similar or even identical optimal solutions, simply because the space of achievable functions is reduced.

Thus, under strong compression constraints, both pruning and distillation may guide the network toward the same limited set of optimal approximations, regardless of the specific method used.

5 Development

Model Architecture and Dataset Integration Framework

The experimental code was constructed using PyTorch deep learning framework with CIFAR-10 dataset. This dataset was chosen because its well-known behavior and simplicity make it easy to work with and computationally inexpensive, allowing us to focus on the core challenges of model compression rather than dataset complexity. The dataset's 32×32 pixel images across 10 classes offered an optimal balance for investigating compression effects without requiring prohibitive computational resources.

A Teacher Model was implemented using the ResNet18 architecture to ensure the network was sufficiently over-parameterized for meaningful compression experiments while maintaining strong baseline performance. This architecture includes standard convolutional layers, batch normalization, and residual connections, making it a representative example of modern neural networks for compression analysis. The model was designed to be modular and extensible, allowing easy integration of different compression techniques while keeping the architecture consistent across all experiments.

The dataset integration framework incorporated simple data preprocessing, such as normalization and train-test partitioning to ensure valid evaluation. Data loading was optimized for the available computational resources by using memory-efficient batch processing.

Modular Code Architecture Development

The codebase was built using software engineering best practices, with a modular design that made it easy to develop, test, and combine different compression techniques. This structure allowed us to study each compression method separately and also to integrate them for experiments involving hybrid approaches.

- **model.py**: Contains the definitions for the Teacher and Student neural network models based on a modified ResNet18 architecture for CIFAR-10. This module includes utilities to adapt the architecture for small images.
- **pruning.py**: Implements both structured and unstructured pruning methods for neural networks, including

global magnitude-based pruning and channel/weight pruning for convolutional and linear layers. The module also provides utilities for freezing pruned weights, iterative pruning with fine-tuning, and loading or reconstructing pruned models, extending PyTorch's pruning functionality to support advanced experimental workflows.

- **distillation.py**: Advanced knowledge distillation framework implementing temperature-scaled softmax operations, custom loss function combinations, and progressive student model generation protocols. The framework supported various distillation strategies and student-teacher architectural configurations.
- **utils.py**: Utility functions containing data loading protocols, training loop implementations, evaluation metrics calculation, model checkpointing systems, and visualization frameworks. This module provided the infrastructure to support all experimental operations.
- **config.py**: Centralized configuration management system providing parameter specification, experimental reproducibility controls, and systematic hyperparameter organization. The configuration system secured consistent conditions across all experimental setups.
- **main.py**: Main experimental pipeline that orchestrates the training, pruning, distillation, and evaluation of models. This script manages the workflow for generating, compressing, and assessing models, as well as saving results and visualizations.
- **activation.py**: Module dedicated to advanced analysis and visualization of internal model representations. It includes tools for extracting activations, computing representational similarity (CKA), comparing models across compression levels, and generating heatmaps and trend plots for interpretability.

This modular architecture made it easy to develop and test each compression technique separately, while also allowing them to be combined for hybrid experiments. The design was essential for managing the complexity of the project and ensured that the code remained easy to maintain and extend.

Baseline Model Establishment

To ensure reliable benchmarks for evaluating compression techniques, we first established a strong baseline model. This involved training a teacher model using standard protocols: we set random seeds for reproducibility, optimized with cross-entropy loss to achieve the highest possible test accuracy, and saved checkpoints of both the initial and fully trained model weights. Throughout this process, we tracked key metrics such as accuracy and parameter count, providing a solid reference point for comparing the effects of pruning and distillation. These steps ensured that all subsequent experiments could be measured against a consistent and statistically validated baseline.

STANDALONE STARTING POINT

The primary objective at this stage was to implement all the core functions and methods required to apply the two main compression strategies—pruning and knowledge distillation—in a standalone manner. This foundational work was essential to ensure that each technique was working as expected in isolation, so that later on when going into deeper stages of the project.

Pruning Implementation:

We developed both structured and unstructured pruning routines. Structured pruning targeted entire channels or filters, which is more compatible with hardware acceleration, while unstructured pruning removed individual weights based on magnitude. The pruning module was designed to be flexible, supporting various pruning ratios and strategies.

The initial approach utilized PyTorch's built-in pruning methods, which allow for the removal of weights or entire channels based on magnitude criteria. However, during early experiments, we observed that the default pruning routines did not fully prevent updates to pruned (zeroed) weights during subsequent training. This was due to the fact that PyTorch's pruning reparameterization leaves the pruned weights in the computational graph, allowing their gradients to be updated unless additional steps are taken.

To address this, we developed custom pruning routines that not only apply the desired pruning mask but also ensure that pruned weights remain fixed throughout fine-tuning. Specifically, after pruning, we register backward hooks on the weight tensors to zero out gradients corresponding to pruned weights, effectively freezing them. This guarantees that only the remaining (unpruned) parameters are updated during optimization, preserving the intended sparsity pattern.

Our pruning module supports both:

- Structured pruning: Removal of entire output channels in convolutional layers or neurons in linear layers, which is more hardware-friendly.
- Unstructured pruning: Removal of individual weights based on global or layer-wise magnitude.

After initial experiments and due to time constraints, the main results and implementations focused on structured pruning. This approach was easier to use and, according to previous research, was best suited to the resources available for this project.

After each pruning step, the model is fine-tuned for a fixed number of epochs to recover any lost accuracy. These included adaptive learning rate schedules and early stopping criteria to help recover accuracy lost due to parameter removal. The pruned models are saved at each compression level for further analysis. This robust and reproducible pruning pipeline was essential for generating reliable baselines and for subsequent experiments involving hybrid compression strategies.

Knowledge Distillation Implementation:

For knowledge distillation, we implemented a teacher-student training framework. The teacher model, typically a larger and fully trained network, provided soft targets for the student model via temperature-scaled softmax outputs. The distillation loss was formulated as a weighted sum

of the standard cross-entropy loss (using ground truth labels) and the Kullback-Leibler divergence between the teacher and student outputs. The framework allowed for configurable temperature and loss weighting parameters, allowing systematic exploration of their effects on student performance.

The standalone distillation approach used, as the student architecture, the model obtained after a pruning process. This ensures that the distilled student model being compared to a pruned model has the same architecture and number of parameters. For this reason, we also saved the random initialization weights of the original model. When loading a pruned model from a *'.pth'* file containing its trained weights, we reset the weights that are not frozen back to their initial values from the original model. This guarantees a fair comparison between pruned and distilled models with identical architectures and parameter counts.

Utility and Infrastructure Functions:
To support these core compression methods, we implemented a set of utility functions. These included model initialization routines (ensuring consistent random seeds and reproducibility), model state management, evaluation metrics (such as accuracy and parameter count), and data loading pipelines optimized for efficient batch processing. This infrastructure was a key point for maintaining experimental consistency and enabling large-scale, automated experimentation with minimal changes.

Rationale for Modular Design:
By developing pruning and distillation as independent modules, we established a clear experimental baseline for each technique. This modular approach not only facilitated direct comparison between methods but also set the groundwork for subsequent phases, where hybrid and sequential compression strategies were explored. The standalone implementations served as reference points, ensuring that any observed effects in combined approaches could be attributed to genuine interactions between techniques rather than confounding implementation details.

HYBRID COMPRESSION

After validating the standalone pruning and distillation modules, we implemented a hybrid compression framework to explore the sequential combination of these techniques. The core of this approach is a distillation-first branching strategy, designed to map the performance and representational effects of applying knowledge distillation followed by further pruning.

Distillation at Multiple Compression Levels:
For a set of target compression levels (expressed as fractions of remaining parameters), we first load the corresponding pruned model checkpoint architecture. Each pruned model is reinitialized with the original random weights for unfrozen parameters to ensure fair comparison. Knowledge distillation is then performed using the trained teacher model as reference, with the student model (matching the pruned architecture) trained for a fixed number of epochs using a temperature-scaled softmax and a combined loss function. The accuracy of each distilled model is evaluated and recorded, forming the main "distillation path" in our experimental landscape.

Branching Pruning from Distilled Models:
From each distilled checkpoint, we further apply structured pruning to reach all higher compression levels (i.e., lower parameter retention). For each branch:

- A deep copy of the distilled model is created to ensure independence of each experimental branch.
- Additional pruning is applied to reach the next target compression level, followed by fine-tuning.
- The accuracy of each pruned-from-distilled model is evaluated and recorded.
- This process is repeated recursively, creating a tree-like structure where each branch represents a unique sequence of distillation followed by further pruning.

Visualization of Hybrid Compression Paths:
To interpret the results, we visualize the performance landscape using dual-axis plots:

- The main distillation path is plotted as a blue line, showing the accuracy of models distilled at each compression level.
- Pruning branches originating from each distilled checkpoint are plotted as red lines, illustrating the effect of further pruning on already distilled models.
- A secondary x-axis is included to provide both the distillation and pruning perspectives on the same plot.

By plotting this visualization, we can clearly observe how the process affects model behavior across all compression levels, revealing trends and patterns that might not be apparent from raw data alone.

EVALUATION AND ANALYSIS PIPELINE

Performance Metrics and Tracking
At each stage of the compression experiments, we recorded key performance metrics, including test accuracy and parameter count, for all pruned, distilled, and hybrid models. This enabled direct comparison of the effectiveness of each compression strategy and provided a quantitative basis for the representational analysis.

Layer-wise Representational Similarity Analysis
To investigate how compression techniques affect internal representations, we implemented a representational similarity analysis using Centered Kernel Alignment (CKA). For each model, activations were extracted from all relevant layers in response to a fixed set of test inputs.

Pairwise CKA similarity was computed between all models at each compression level, enabling four primary comparison types: pruned versus distilled, mixed versus pruned, mixed versus distilled, and mixed versus mixed. This systematic approach allowed us to map representational convergence and divergence across the entire experimental matrix.

Orthogonality Assessment

To quantify the independence or overlap between pruning and distillation, we developed an orthogonality assessment framework based on CKA similarity patterns. For each compression level, we aggregated CKA scores for each comparison type and evaluated whether the observed similarities indicated orthogonal, redundant, or complementary effects. Statistical validation was performed using confidence intervals and variance analysis to ensure the robustness of our findings.

Automated Visualization and Data Management

The analysis pipeline included automated generation of visualizations and structured data outputs. Key visualizations included:

- **Comprehensive CKA similarity heatmaps** summarizing representational similarity across all model pairs and compression levels.
- **Trend plots** showing how average CKA similarity for each comparison type evolves as a function of compression intensity.
- **Layer-wise similarity heatmaps and boxplots** to highlight which network components are most affected by each compression method.

All results were saved with timestamp-based file management for reproducibility and traceability.

Pipeline Integration

The evaluation and analysis pipeline was fully integrated with the experimental framework, enabling automated model discovery, loading, and comparison. This ensured that all models generated during the hybrid compression experiments could be systematically analyzed without manual intervention. The resulting outputs and visualizations provided the necessary data to interpret the effects of compression strategies on both performance and internal representations.

This evaluation framework enabled us to address the central research question: whether different compression techniques discover similar or distinct representational solutions, and under what conditions their effects are complementary or redundant.

6 Results

6.1 Compression-Level Dependent Representational Convergence

Our analysis demonstrates a strong relationship between network capacity and representational similarity across compression techniques.
As shown in Table 1, the average CKA similarity between pruned and distilled models increases from 0.7004 at 10% capacity to 0.8592 at 90% capacity, indicating a progressive convergence as compression pressure decreases. This trend is consistent across all types of comparisons, including mixed models.

Compression Level	CKA Similarity	Relationship
10%	0.700	Technique-specific solutions
20%	0.788	Moderate divergence
30%	0.810	Increasing similarity
40%	0.806	Transitional convergence
50%	0.836	Progressive similarity
60%	0.837	Strong convergence
70%	0.840	High similarity
80%	0.858	Near-complete convergence
90%	0.859	Maximum convergence

TABLE 1: Progressive convergence pattern for pruned vs distilled models.

At the lowest capacity (10%), the CKA similarity between pruned and distilled models is at its minimum (0.7004), indicating that the two techniques yield the most distinct internal representations under extreme compression. As capacity increases, this divergence diminishes, and by 90% capacity, the similarity reaches 0.8592, reflecting a strong convergence of learned representations. This progression highlights how resource constraints drive the diversity or overlap of solutions found by different compression strategies.

To provide a clear overview of how representational similarity evolves across all comparison types and compression levels, we summarize the results in the following comprehensive heatmap.

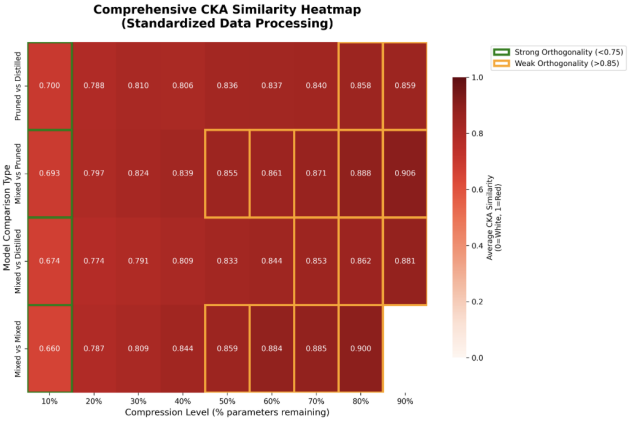

Fig. 5: CKA similarity heatmap for all model pairs and compression levels.

Each cell shows the average CKA similarity for a given comparison type (rows) and compression level (columns). Green borders indicate strong orthogonality (low similarity, CKA < 0.70), while orange borders highlight weak orthogonality or high similarity (CKA > 0.85). The plot demonstrates that as more parameters are retained (moving right), the similarity between models increases, indicating that pruning and distillation produce more similar representations at higher capacities.

- **Pruned vs Distilled**: Average CKA = 0.815
- **Mixed vs Pruned**: Average CKA = 0.837
- **Mixed vs Distilled**: Average CKA = 0.814
- **Mixed vs Mixed**: Average CKA = 0.829

To further illustrate how representational similarity evolves as a function of compression, we present the following trend plot, which tracks the average CKA similarity across all comparison types and compression levels.

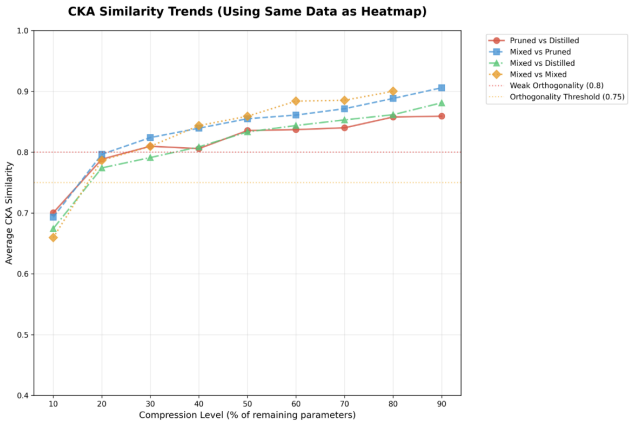

Fig. 6: CKA similarity trends across compression levels for all comparison types.

The lines in the plot represent the average CKA similarity between the different model pairs at each compression level. For mixed models to provide additional benefits—such as learning complementary or more diverse representations—their similarity to pruned or distilled models should be lower, indicating that they capture different aspects of the data. However, the observed trends show that mixed models have CKA similarity values comparable to, or even higher than, those of pruned vs distilled comparisons. This suggests that combining pruning and distillation does not lead to more diverse internal representations; instead, the resulting models largely converge to similar solutions as those produced by each technique alone.

6.2 Layer-Wise Representational Architecture

Layer-wise CKA analysis reveals that early layers (e.g., conv1, bn1) exhibit high similarity (CKA $\approx$ 0.97–0.99) across all methods and compression levels, indicating universal convergence in basic feature extraction. In contrast, deeper layers, particularly the second convolution in each ResNet block, show greater differentiation, with CKA values dropping as low as $\approx$ 0.30–0.65, especially at high compression values (e.g., 10% of weights remaining). This differentiation is more pronounced at lower compression levels.

ResNet Block Patterns:

- **First convolution:** Moderate differentiation (CKA $\approx$ 0.97–0.99)
- **Second convolution:** Maximum differentiation (CKA $\approx$ 0.30–0.65)
- **Batch normalization:** Higher similarity than convolutions (CKA $\approx$ 0.97–0.98 in early layers, but can drop to $\approx$ 0.30–0.64 in deeper layers)
- **Skip connections:** Intermediate (CKA $\approx$ 0.75–0.85, depending on layer depth and compression)

To visualize these patterns, we generated heatmaps that display layer-wise similarity using four different similarity measures.

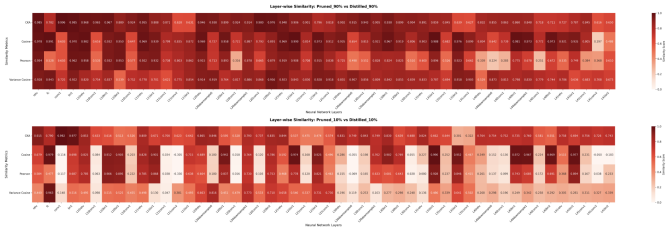

Fig. 7: Top: Similarity heatmap between layers at 90% capacity. Bottom: Similarity heatmap at 10% capacity.

The heatmaps above display four types of similarity measures (Y axis) across all layers of the models (X axis), comparing pruned and distilled networks. As indicated by the color gradients, models retaining 90% of their parameters exhibit high similarity across layers, whereas models compressed to 10% of their original size show lower similarity values.

By examining the plots, we can see that as the layers become deeper (moving to the right on the X-axis), the similarity values decrease, which is reflected by lighter colors in the plot that become more common on later layers even in the case of 10% capacity.

This pattern highlights that representational convergence between pruning and distillation is strongest at higher capacities and diminishes as compression becomes more extreme.

6.3 Performance Trends Across Compression Levels

To complement the representational similarity analysis, we evaluated the test accuracy of models distilled and pruned at matching compression levels. Figure 8 shows the accuracy curves for models subjected to distillation followed by pruning, across a range of pruning percentages.

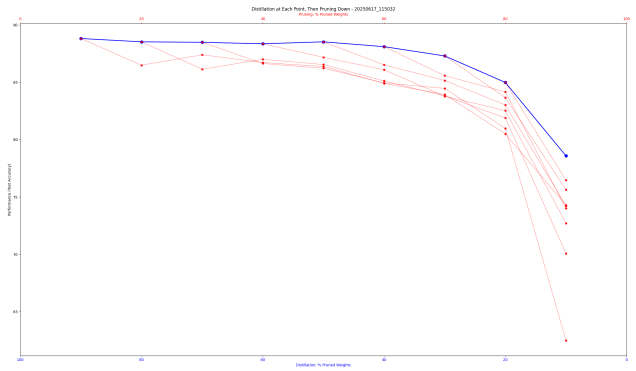

Fig. 8: Test accuracy of models distilled and pruned at the same compression levels. The blue curve shows the mean accuracy, while red lines indicate individual runs.

The performance curves for models distilled and pruned at the same levels closely follow each other, indicating that

both compression techniques not only produce similar internal representations but also result in comparable predictive performance across the entire compression spectrum.

6.4 Compression-Dependent Subnetwork Discovery

Our analysis reveals that the degree of representational overlap between pruned and distilled models is strongly dependent on the level of compression applied. Specifically, three distinct regimes can be identified:

- **Extreme Compression (10–30% parameters retained):** CKA values range from 0.65 to 0.82, indicating that each technique tends to discover its own, largely non-overlapping subnetwork. This suggests that under severe resource constraints, the optimization landscape allows for more diverse solutions, and the effects of pruning and distillation are less aligned.
- **Moderate Compression (40–60% parameters retained):** CKA values increase to 0.81–0.88, marking a transitional state. Here, subnetworks discovered by each method begin to overlap, and representational convergence becomes more apparent, though some technique-specific differences remain.
- **Mild Compression (70–90% parameters retained):** CKA values further rise to > 0.85, reflecting a high degree of similarity and overlap in the subnetworks identified by pruning and distillation. In this regime, the optimization landscape appears to constrain both methods toward similar, near-optimal solutions.

Overall, these results highlight that as network capacity increases, the subnetworks identified by pruning and distillation become more similar. This convergence suggests that, with more available parameters, both techniques are guided toward overlapping or even nearly identical solutions, while at higher compression levels, their effects remain more distinct.

7 Discussion

Our results reveal a clear pattern: **as network capacity increases, the internal representations of pruned and distilled models become more similar**. This is evident in the progressive increase in CKA similarity and is visually supported by the heatmaps and performance curves presented earlier. Such convergence aligns with the Manifold Hypothesis, which suggests that neural networks, regardless of the compression method, tend to organize data into similar geometric structures when given sufficient capacity. In other words, when models are less constrained, both pruning and distillation appear to guide the network toward comparable solutions.

This trend is further supported by the Information Bottleneck Theory. **Both pruning and distillation encourage the network to compress information efficiently**, and as the available capacity grows, their effects begin to overlap. The performance results reinforce this, showing that models compressed by either method achieve nearly identical accuracy across most compression levels. While the two techniques differ in their approach, their impact on the network's information processing becomes increasingly similar as more parameters are retained.

A closer look at the layer-wise analysis reveals that **early layers, responsible for basic feature extraction, remain highly similar across all methods and compression levels**. However, differences become more pronounced in deeper layers, especially under high compression. This suggests that, when resources are limited, pruning and distillation may lead to more method-specific solutions in the deeper parts of the network. As capacity increases, the subnetworks identified by both techniques begin to overlap, supporting the Lottery Ticket Hypothesis, which posits that optimal subnetworks can be found by different ways when the model is not too restricted.

Mixed models do not provide more diverse or complementary representations than pruning or distillation alone. In the context of representational similarity, lower CKA values between mixed models and either pruned or distilled models would indicate that mixed models are learning more diverse or complementary representations. However, as shown in our results, the average CKA similarity for mixed models remains high and closely tracks the values observed for pruned vs distilled comparisons. This suggests that combining pruning and distillation does not lead to the discovery of fundamentally different or more diverse internal representations. Instead, the mixed models largely converge to similar solutions as those produced by each technique alone. Consequently, from a representational perspective, mixed models do not provide additional functional benefits over using pruning or distillation individually, except perhaps under the most extreme compression where some divergence may still occur.

From a practical perspective, these findings indicate that **for moderate and mild compression, the choice between pruning and distillation may not be critical**—both methods obtain similar internal representations and predictive performance. This flexibility is valuable for practitioners as it allows them to select the approach that best fits their workflow or computational constraints. However, under extreme compression, the differences between methods become more significant, and careful selection or combination of techniques may be necessary to achieve optimal results.

It is important to acknowledge the limitations of this study. Our experiments were conducted on the CIFAR-10 dataset using a ResNet-based architecture. Future research should investigate whether these patterns hold for larger datasets and different model types, such as transformers or domain-specific architectures. Additionally, exploring how these findings translate to other tasks, like transfer learning or robustness to adversarial attacks, would be valuable. Finally, the development of adaptive or hybrid compression strategies that leverage the observed convergence could further enhance model efficiency and generalization.

Overall, our findings provide new insights into the interplay between pruning and knowledge distillation, clarifying

when their effects are complementary and when they are redundant, and offering practical guidance for the design of efficient neural networks.

8 Conclusions

This thesis investigated the relationship between two of the most widely used neural network compression techniques: pruning and knowledge distillation. By combining an experimental design with advanced representational analysis, we set out to determine whether these methods act orthogonally, redundantly, or interfering when applied to modern deep learning models.

Our results reveal that the effects of pruning and distillation are not fully orthogonal. Instead, their impact on both model performance and internal representations depends strongly on the degree of compression. At high compression levels (i.e., when only a small fraction of parameters are retained), pruning and distillation tend to discover distinct, technique-specific subnetworks, resulting in lower representational similarity and more pronounced differences in accuracy. However, as network capacity increases, the representations and performance of pruned and distilled models converge, indicating that both methods are guided toward similar solutions by the underlying optimization landscape.

Layer-wise analysis further showed that early layers are robust to compression and remain highly similar across methods, while deeper layers are more sensitive and reflect the main differences between techniques, especially under extreme compression. These findings are consistent with theoretical frameworks such as the Manifold Hypothesis, the Information Bottleneck Principle, and the Lottery Ticket Hypothesis, all of which suggest that the solution space narrows as model capacity increases, promoting convergence between different compression strategies.

From a practical point of view, our work suggests that for moderate and mild compression, practitioners can choose either pruning or distillation—or even combine them—without significant loss in performance or representational diversity. Only in the regime of extreme compression does the choice of method become critical, as the differences between approaches become more pronounced.

In summary, this thesis provides new insights into the connection between pruning and knowledge distillation, clarifying when their effects are complementary and when they are redundant. These findings can inform the design of more efficient and robust neural networks, especially in resource-constrained environments.

Acknowledgments

I would like to express my deepest gratitude to my thesis supervisor, Andrey Barsky, for his invaluable guidance, support, and encouragement throughout this project. His expertise and insightful feedback have been essential to the development and completion of this work.

I am also profoundly grateful to my family for their unconditional support, patience, and understanding during my studies. Their encouragement has been a constant source of motivation and strength.

Thank you all for making this journey possible.

As we attempt to advance artificial intelligence, let us remember that true progress lies not only in building ever larger models, but in learning to optimize and refine what we already have. Sustainable and efficient AI will be achieved by those who master the art of doing more with less.

REFERENCES

[1] Aghli, N., & Ribeiro, E. (2021). Combining weight pruning and knowledge distillation for CNN compression. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 3185–3192. https://doi.org/10.1109/CVPRW53098.2021.00356

[2] Bansal, Y., Nakkiran, P., & Barak, B. (2021). Revisiting model stitching to compare neural representations. In arXiv [cs.LG]. http://arxiv.org/abs/2106.07682

[3] Kim, J., Chang, S., & Kwak, N. (2021). PQK: Model compression via pruning, quantization, and knowledge distillation. In arXiv [cs.LG]. http://arxiv.org/abs/2106.14681

[4] Muralidharan, S., Sreenivas, S. T., Joshi, R., Chochowski, M., Patwary, M., Shoeybi, M., Catanzaro, B., Kautz, J., & Molchanov, P. (2024). Compact language models via pruning and knowledge distillation. Neural Information Processing Systems, abs/2407.14679. https://doi.org/10.48550/arXiv.2407.14679

[5] Pan, B., Jiao, J., Pang, J., & Cheng, J. (2024). Distill-then-prune: An efficient compression framework for real-time stereo matching network on edge devices. In arXiv [cs.CV]. http://arxiv.org/abs/2405.11809

[6] Park, J., & No, A. (2021). Prune your model before distill it. In arXiv [cs.LG]. http://arxiv.org/abs/2109.14960

[7] Sariyildiz, M. B., Weinzaepfel, P., Lucas, T., Larlus, D., & Kalantidis, Y. (2024). UNIC: Universal classification models via multi-teacher distillation. In arXiv [cs.CV]. http://arxiv.org/abs/2408.05088

[8] Bäuerle, A., Jönsson, D., & Ropinski, T. (2022). Neural activation patterns (NAPs): Visual explainability of learned concepts. In arXiv [cs.LG]. http://arxiv.org/abs/2206.10611

[9] Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., & Belilovsky, E. (2022). Reliability of CKA as a similarity measure in deep learning. In arXiv [cs.LG]. http://arxiv.org/abs/2210.16156

[10] Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. In arXiv [cs.LG]. http://arxiv.org/abs/1905.00414

[11] Anokhin, I., & Yarotsky, D. (2020). Low-loss connection of weight vectors: distribution-based approaches. International Conference on Machine Learning, 335–344. https://proceedings.mlr.press/v119/anokhin20a/anokhin20a.pdf

[12] Augustine, M. T. (2024). A Survey on Universal Approximation Theorems. In arXiv [cs.LG]. http://arxiv.org/abs/2407.12895

[13] Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In arXiv [cs.LG]. http://arxiv.org/abs/1803.03635

[14] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. In arXiv [stat.ML]. http://arxiv.org/abs/1503.02531

[15] LeCun, Y., Denker, J., & Solla, S. (1989). Optimal Brain Damage. Neural Information Processing Systems, 598–605. https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf

[16] Lee, J., Hwang, S.-W., Qiao, A., Campos, D. F., Yao, Z., & He, Y. (2024). STUN: Structured-then-unstructured pruning for scalable MoE pruning. In arXiv [cs.LG]. http://arxiv.org/abs/2409.06211

[17] Liu, Y., Zhang, W., & Wang, J. (2021). Adaptive Multi-Teacher Multi-level Knowledge Distillation. In arXiv [cs.CV]. http://arxiv.org/abs/2103.04062

[18] Park, S. (2024). Diverse Feature Learning by self-distillation and reset. In arXiv [cs.AI]. http://arxiv.org/abs/2403.19941

[19] Şimşek, B., Ged, F., Jacot, A., Spadaro, F., Hongler, C., Gerstner, W., & Brea, J. (2021). Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In arXiv [cs.LG]. http://arxiv.org/abs/2105.12221

[20] Tishby, N., & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In arXiv [cs.LG]. http://arxiv.org/abs/1503.02406

[21] Whiteley, N., Gray, A., & Rubin-Delanchy, P. (2022). Statistical exploration of the Manifold Hypothesis. In arXiv [stat.ME]. http://arxiv.org/abs/2208.11665

[22] Zhu, M., & Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. In arXiv [stat.ML]. http://arxiv.org/abs/1710.01878

[23] Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. In arXiv [cs.NE]. http://arxiv.org/abs/1506.02626

[24] Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning Filters for Efficient ConvNets. In arXiv [cs.CV]. http://arxiv.org/abs/1608.08710

[25] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In arXiv [cs.CV]. http://arxiv.org/abs/1708.06519

[26] Yim, J., Joo, D., Bae, J., & Kim, J. (2017). A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. Proceedings of the CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 7130–7138. https://doi.org/10.1109/CVPR.2017.754

[27] Kim, K., Ji, B., Yoon, D., & Hwang, S. (2020). Self-knowledge distillation with progressive refinement of targets. In arXiv [cs.LG]. http://arxiv.org/abs/2006.12000